

REMARKS

This is a full and timely response to the outstanding non-final Office Action mailed April 1, 2009. Reconsideration and allowance of the application and pending claims are respectfully requested.

Claim Rejections - 35 U.S.C. § 103(a)

Claims 1, 2, 5, 9, 13, 15, 25, 26, 28-30, and 35-39 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over *Karakashian, et al.* ("Karakashian," U.S. Pub. No. 2004/0064503) in view of *Felciano, et al.* ("Felciano," U.S. Pat. No. 6,052,730). Applicant respectfully traverses.

As has been acknowledged by the Court of Appeals for the Federal Circuit, the U.S. Patent and Trademark Office ("USPTO") has the burden 35 U.S.C. § 103 to establish obviousness by showing objective teachings in the prior art or generally available knowledge of one of ordinary skill in the art that would lead that individual to the claimed invention. *In re Fine*, 837 F.2d 1071, 1074, 5 U.S.P.Q. 2d 1596, 1598 (Fed. Cir. 1988). The key to supporting an allegation of obviousness under 35 U.S.C. § 103 is the clear articulation of the reasons why the Examiner believes that claimed invention would have been obvious. See MPEP § 2141. As stated by the Supreme Court, "[r]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *KSR v. Teleflex*, 550 U.S. at 398, 82 USPQ2d 1385 (quoting *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006)). Applicant respectfully submits that the Examiner has not

established with clearly articulated reasons that Applicant's claims are obvious in view of the prior art. Applicant discusses those claims below.

A. Claims 1, 2, 5, 9, 13, and 15

Applicant's independent claim 1 provides as follows:

1. A method for building a session timing profile, the method comprising:
 - a client sending a request intended for a network service;
 - a message handler associated with the client intercepting the request;
 - the message handler interjecting a session identifier into the request;
 - the message handler transmitting the request to the network service over a network;
 - the message handler storing in a database relative to the session identifier the time at which the request was transmitted to the network service;
 - the message handler intercepting a response to the request from the network service and intended for the client;
 - the message handler identifying the session identifier within the response;
 - the message handler storing in the database relative to the session identifier the time at which the response was received; and
 - the message handler providing the response to the client.

In the Office Action, it is argued that Karakashian discloses each of the above limitations with the exception of a message handler storing the time at which a request was transmitted to a network service or storing in the database relative to a session

identifier the time at which a response was received. While Applicant agrees that Karakashian fails to disclose or suggest those aspects of claim 1, Applicant disagrees that Karakashian discloses the other limitations of the claim. Applicant discusses some of those other limitations below.

1. A Message Handler “Associated with” a Client Intercepting a Request Intended for a Network Service

As a first matter, Karakashian does not in fact disclose “a message handler associated with the client intercepting the request” intended for a network service, as recited in claim 1. In the Office Action it is argued that Karakashian discloses such intercepting in paragraph 0032, lines 4-7. That paragraph provides as follows:

FIG. 1 shows the relationship of a web container 108 and SMTP listener 104 and a host server or web service container 108, utilizing an architecture in accordance with one embodiment of the present invention. A protocol adapter for HTTP 102 is shown in a web container 100, which can intercept a web service invoke via HTTP from a web services client. A protocol adapter for SMTP 106 is also shown in an SMTP listener 104, which can accept a web service invoke via SMTP. This architecture allows for pluggability in a number of places.

Karakashian, paragraph 0032. As can be appreciated from the above excerpt, Karakashian describes a web container 100 *associated with* a web service container 108 that intercepts a web service invoke via HTTP. Nowhere does Karakashian indicate that the web container 100 is “associated with” the client that sent the invoke. Indeed, it appears clear that the web container 100 has no association whatsoever with the client.

Instead, the web container 100 merely receives the invoke transmitted over the Internet. Moreover, nothing in the Karakashian disclosure or other identified prior art provides a suggestion to modify Karakashian's system to associate the web container 100 with the client. Therefore, such a modification would not have been obvious to a person having ordinary skill in the art.

In the Response to Arguments section of the Office Action, it is argued that Karakashian discloses a message handler associated with the client intercepting the request in paragraph 0036. Turning to that paragraph, Karakashian discloses:

A protocol adapter can be responsible for identifying requests as web service messages, as well as routing the messages to a web services container. If the protocol being used supports synchronous responses, a protocol adapter can also receive the response data and return the data to the originator of the request. The protocol adapter can convert the message to the original message format if it is not SOAP plus attachments. A protocol adapter can deal with any message context that is required by the container, such as a conversation ID, and is transmitted at the protocol level, such as cookies in HTTP. The protocol adapter can propagate the message context to and from a web services container.

Karakashian, paragraph 0036. As with paragraph 0032, it is clear that paragraph 0036 does not in fact describe a message handler "associated with" the client intercepting the request intended for a network service. Instead, Karakashian describes a protocol adapter 102, which is part of the web container 100, receiving and routing requests. As identified above, the web container 100 is in no way associated with the client that sent in the request. To the contrary, the web container 100 is clearly associated with the

web service container 108 to which the request is intended. See *Karakashian*, Figure 1.

2. The Message Handler Interjecting a Session Identifier into the Request

Karakashian also does not actually disclose the message handler “interjecting a session identifier into the request”, as recited in claim 1. In the Office Action it is argued that Karakashian describes such interjecting in paragraphs 0036 and 0038. Paragraph 0036 has been reproduced above. Paragraph 0038 provides as follows:

An invocation context can be used, which is an inheritable thread-local object that can store arbitrary context data used in processing a web service request. The context can be available from various components of the architecture involved in the processing of the request and response. Typical data that might be stored in such a context are a conversation ID, a message sequence number, and a security token. A particular invocation handler can choose to make the invocation context available to the target component. This can allow application code to read and write to the invocation context.

Karakashian, paragraph 0038.

Beginning with paragraph 0036, Karakashian does not actually disclose the message handler “interjecting a session identifier into the request” in that portion of his disclosure. Indeed, as can be readily appreciated from paragraph 0036 (see above), Karakashian says nothing whatsoever of a message handler, or any other component for that matter, interjecting a session into a request. Instead, Karakashian simply

indicates that the protocol adapter 102 identifies requests and routes them to the web services container 108. Although the protocol adapter 102 is described as being able to “convert” the message format, that action does not inherently include interjecting a session identifier into the message. Furthermore, although the protocol adapter 102 is described as being able to “deal with” a conversation ID, Karakashian does *not* indicate that the protocol adapter interjects that ID into the request.

Regarding Karakashian’s paragraph 0038, Karakashian is similarly silent as to a message handler interjecting a session identifier. Instead of disclosing such interjection, paragraph 0038 simply describes an invocation context in which a thread-local object can store arbitrary context data. Although the conversation ID can be stored *in the thread-local variable*, Karakashian does not indicate that the ID is interjected into the request.

3. The Message Handler Intercepting a Response to the Request from the Network Service Intended for the Client

As a further matter, Karakashian does not in fact disclose the message handler “intercepting a response to the request from the network service and intended for the client”, as provided in claim 1. In addressing that limitation, the Office Action cites paragraph 0036 of the Karakashian disclosure, which has been reproduced above.

As before, paragraph 0036 does not contain the purported disclosure. Instead of disclosing “intercepting” a response intended for the client, Karakashian merely indicates that the protocol adapter 102 “can also receive” a response. Therefore, Karakashian does not indicate that the protocol adapter 102 “intercepts” the response

which was sent by the web service container 108 to the client. Such “interception,” which inherently identifies receiving something intended for another target, is not necessary given that the web service container 108 appears to send the response directly to the web container 100 and its protocol adapter 102.

4. The Message Handler Intercepting a Response to the Request from the Network Service Intended for the Client

As described above, it is acknowledged in the Office Action that Karakashian does not disclose “the message handler storing in a database relative to the session identifier the time at which the request was transmitted to the network service”. In view of that shortcoming, the Office Action cites column 4, lines 51-65 of the Felciano reference. That portion of Felciano’s disclosure provides:

In the preferred embodiment, the lamprey program maintains a database of log files which store the client HTTP request information. Each time a request passes through lamprey, the following information is logged in a tab-delimited text file:

1. User ID (entered by the user)
2. Date & time stamp
3. URL (link destination)
4. Date & time stamp for when the source page was generated
5. Referrer
6. IP address of client
7. Hostname of client
8. User.sub.-- agent (client web browser name)

Felciano, column 4, lines 51-65. Although *Felciano* describes various information associated with a client request being stored, *Felciano* does not identify any message handler associated with the client that performs such actions. In *Felciano*'s system, the CGI script program "Lamprey," which comprises part of an HTTP server (see *Felciano*, Figure 1), receives the requests and stores the relevant information. Clearly, such a program cannot reasonably be considered to be "associated with" the client that sent the request.

5. The Message Handler Storing in the Database Relative to the Session Identifier the Time at which the Response was Received

As was also described above, it is acknowledged in the Office Action that Karakashian does not disclose "the message handler storing in the database relative to the session identifier the time at which the response was received". In view of that shortcoming, the Office Action cites column 4, lines 51-65 of the *Felciano* reference, which was reproduced above.

As described above, *Felciano* fails to disclose or suggest a message handler "associated with" the client storing any information regarding a received message. As a further matter, Applicant notes that *Felciano* more generally fails to disclose or suggest storing information regarding a *response* to a received request. Instead, *Felciano*'s disclosure is focused on storing information concerning requests, not responses to those requests.

6. Conclusion

In view of the foregoing, it is clear that Karakashian and Felciano do not disclose or suggest each of Applicant's claim limitations. In view of that fact, Applicant respectfully requests that the rejections be withdrawn.

B. Claims 25, 26, 28-30, and 35

Applicant's independent claim 25 provides as follows:

25. A computer-readable medium that stores a message handler associated with a client, the message handler comprising:

 logic configured to intercept a message sent by a client and intended for a network service;

 logic configured to interject a session identifier into the message;

 logic configured to transmit the message to the network service via a network; and

 logic configured to store in a database relative to the session identifier the time at which the message was transmitted to the network service.

Regarding claim 25, neither Karakashian nor Felciano disclose or suggest a message handler "associated with a client" that includes "logic configured to intercept a message sent by the client and intended for a network service", "logic configured to interject a session identifier into the message", or "logic configured to store in a database relative to the session identifier the time at which the message was transmitted to the network service", at least for reasons described above in relation to claim 1. In view of that fact, Applicant respectfully requests that the rejections be withdrawn.

C. Claims 36-39

Applicant's independent claim 36 provides as follows:

36. A computer-readable medium that stores a message handler associated with a network service, the message handler comprising:

logic configured to intercept a request sent to the network service from a client;

logic configured to identify a session identifier within the request;

logic configured to store in a database relative to the session identifier the time at which the request was received; and

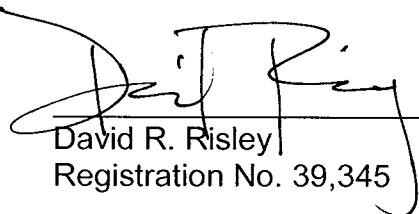
logic configured to provide the request to the network service.

Regarding claim 36, neither Karakashian nor Felciano disclose or suggest a message handler "associated with a client" that includes "logic configured to identify a session identifier within the request" or "logic configured to store in a database relative to the session identifier the time at which the request was received", at least for reasons described above in relation to claim 1. In view of that fact, Applicant respectfully requests that the rejections be withdrawn.

CONCLUSION

Applicant respectfully submits that Applicant's pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,



David R. Risley
Registration No. 39,345